

Preface

The field of artificial intelligence (AI) has made substantial progress in moving from the research laboratory to practical commercial applications. This issue of the *Systems Journal* focuses on those applications of AI known as *knowledge-based systems*. These applications allow computer systems to deliver to users a degree of automatic reasoning. Although such reasoning was difficult to implement and update in past systems using traditional software approaches, tools and techniques now exist to allow a knowledge base to be more easily created and exercised to provide a range of beneficial applications.

The technologies supporting knowledge-based systems are still evolving. Many competing programming languages and environments are available, each providing distinctive solutions and benefits. The papers in this issue discuss a number of these different systems.

The opening paper, by Symonds, provides a concise introduction to the subject and to IBM's products for knowledge-based systems. Symonds also defines the new concepts and terminology associated with these systems. The dynamics of this area are moving swiftly, and new IBM applications continue to pace the field.

Several of the other papers describe applications taken from the disciplines of data processing, including operations management, service diagnostics, programming development, and service cost estimation. These papers reflect a variety of approaches that bring knowledge-based technology to bear on the needs of users. By contrast, the paper by Hagamen and Gardy describes a novel and practical application in a field beyond traditional data processing—that of medical education. It is interesting to note the diversity of professions that are finding value in knowledge-based applications.

Much of the leading research done in artificial intelligence has been accomplished using the LISP programming language—often on specialized workstations. Unfortunately, a wide gulf separates these laboratory systems from commercial data processing environments. The paper by Hodil, Butler, and Richardson describes a knowledge-based application shell system that was developed in PL/I for use in a mainframe environment with access to the terminal network and data bases available on such a system. The authors describe an example application in program development and maintenance to demonstrate the potential of this environment.

As central mainframe computing facilities continue to grow in workload, the tasks of console operators have become more complex. Timely operator responses to system events have become increasingly important because of the dependency of users on dispersed network resources that rely upon the composite availability of many systems. An error in diagnosing a situation or failure to provide the correct response may aggravate the initial problem. The paper by Milliken and his colleagues describes an experimental knowledge-based system to assist with the operation of a large computer complex running the MVS operating system. This paper discusses research on a knowledge base that allows operational strategies and policies to be automated in the computing system.

IBM has announced a knowledge-based systems shell called *Expert Systems Environment* that runs under multiple operating systems. This shell consists of separate development and consulting components. By removing much of the programming burden in creating a knowledge-based application, by providing access to host resources and data bases, and by allowing users to reach applications through networks of general-purpose IBM 3270 displays, Expert

Systems Environment promises to address the needs of a wide range of business applications. It is such applications as these—although not as conspicuous or engaging as running a computing complex or diagnosing an illness—that occupy the days of a large portion of our collective corporate workforces. The paper by Voelker and Ratica describes the evolution of one frequently encountered business application: estimating the labor cost for recurring complex projects in order to support the preparation of proposals and agreements. The authors chronicle their experience, which has resulted in a final production implementation using Expert Systems Environment.

A programmer can usually understand the flow and function of a program in a language foreign to him or her. The programming language *Prolog*, however, presents a particular challenge in this regard. The language statements indicate relationships and provide little clue as to what may be expected to occur when the program is run. The paper by Wilson provides a look at the Prolog language, with the aim of providing an understanding of the form and role of this language, which is frequently used with artificial intelligence applications.

Various techniques for the representation of knowledge have been explored. Many representations have adopted an IF-THEN syntax to define their rules, which, in turn, may be translated into an alternative internal format. The paper by Hagamen and Gardy describes an approach that was effective in relating elements of medical data. The items were represented as integers, and this format was found to be well-suited to fast processing on a workstation using the APL programming language.

The data structures used by knowledge-based systems generally require a large amount of computer memory. The paper by Burns et al. describes a project to use established techniques of demand-memory paging to access knowledge-based segments and thus reduce the amount of real memory needed. This demand-paging technique was implemented in the IBM RT Personal Computer for its diagnostic programming.

As we approach the twenty-first century, a paper on the computation of dates is appropriate. The paper by Ohms presents an interesting history of the calendar and date systems. The author introduces his own *Lilian date format* for computer date representation and computation. Algorithms are discussed that allow for direct conversion of traditional Julian dates to and from this new format.

Knowledge-based systems offer a fresh challenge to the information processing community. Much of the past quarter century of the computer era has been dedicated to automating clerical and manual processes. Now we are building the tools to create a similar revolution in automating and supporting judgmental processes. We hope this issue of the *IBM Systems Journal* provides you with insight into these processes and helps inspire your participation in the effort.

Gary Gershon
Editor